Services and Security

Iowa State University
Information Technology Services

Services and Security

- Find out what services are running
- Find out what's listening to the net
- Find out what they do
- Disable or remove any that you don't need
- References
  http://techrepublic.com.com/5100-6270-1053043.html
  http://linuxplanet.com/linuxplanet/tutorials/211/2

The services in your computer are daemons that provide, well, services to users and programs like remote access, mail and printing. Unfortunately, they also provide channels through which your system can be attacked. Older Linux distributions were often quite insecure out of the box because many insecure services were enabled by default. This is a much smaller problem now, as fewer services are run by default in most distros, but it's still a good idea from both a security and a performance standpoint to shut down any services that your system doesn't actually need.

How are services started?

- From scripts referenced in `/etc/rc.d/rcN.d`
- From xinetd (or, in older distros, from inetd)
  - Xinetd monitors incoming network communications and launches the appropriate processes to handle them

Services are launched in two ways:

- When a runlevel is started, scripts in the `/etc/rc.d/rcN.d` directory beginning with S are run; each of these is associated with a service.
- Xinetd, the Internet superservice, monitors incoming network traffic and, based on packet type, launches an appropriate service to handle it using the scripts in `/etc/xinetd.d`.

What Services are Running?

- `/sbin/chkconfig --list`

```
afs             0:off   1:off   2:off   3:on    4:off   5:on    6:off
irqbalance      0:off   1:off   2:off   3:off   4:off   5:off   6:off
diskacct        0:off   1:off   2:on    3:on    4:on    5:on    6:off
smartd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
load            0:off   1:off   2:on    3:on    4:on    5:on    6:off
xinetd based services:
        chargen-udp:    off
        chargen:        off
        daytime-udp:    off
        daytime:        on
```

  - The numbers refer to runlevels; the services below are started by xinetd
- Try `/sbin/chkconfig --list | grep on`

On a Red Hat system the easy way to display services that are enabled in all runlevels at once is through the command `chkconfig –list`. The resulting table is divided into two parts:

- Regular services, showing each service's status in each of the seven runlevels (on or off).
- Services started by xinetd, showing whether or not the service is enabled. If xinetd is enabled in a runlevel, then all the xinetd-related services can be executed. If xinetd is off, then no network-related service will be launched.

Without Chkconfig...

- Chkconfig is a Red Hat utility and may not be available in other distros (particularly Debian), even if they use System-V type init scripts
- Look in /etc/rc.d/rc.N for scripts beginning with `SNNservicename`
- If you have to do this more than a couple of times...you'll go get the source for chkconfig. It will work under other SysV distros.

If you happen to be running a non-RedHat distribution (like Debian GNU/Linux) you won't have chkconfig available for you to use. In that case, you'll have to examine the /etc/rc.d/rc.N directories for scripts beginning with S; those services are started at runtime. It is possible to compile chkconfig under any distribution that uses System-V styled init scripts.

[Exercise: find out what services are enabled in runlevel 5 on your system using the chkconfig command. If you have time, try it without using chkconfig.]

What are these services?

- Try `man servicename`
- Try `whatis servicename`
- Find and identify the executable
  - For ordinary services, look at the script "/etc/rc.d/init.d/servicename"
  - For xinetd services, look at config file "/etc/xinetd.d/servicename"

Okay, you have a list of service names. How do you find out what they are?

1. Try the man page for the service name.
2. Try the whatis command.
3. If neither of those work, then you have a service created by a lazy and/or inconsiderate programmer who didn't care if people knew what his service was. You'll have to find the executable file from the script that launches the service and interrogate it as to what it is.

/etc/rc.d/init.d/atd

- Look for the "start" section

```
start() {
        # Check if atd is already running
        if [ ! -f /var/lock/subsys/atd ]; then
            echo -n $"Starting $prog: "
            daemon /usr/sbin/atd
            RETVAL=$?
            [ $RETVAL -eq 0 ] && touch /var/lock/subsys/atd
            echo
        fi
        return $RETVAL
}
```

When you look at the script `/etc/rc.d/rcN.d/SNNservicename`, you'll find it's a symbolic link to a script in the `/etc/rc.d/init.d` directory. If you're lucky, this script will have the same name as the service, and the script will be commented so you can simply read what the service does. The location of the executable will be in the section beginning with “start()” in a line beginning with the word “daemon”.

If you're not so lucky, you may have to examine the `/etc/rc.d/rcN.d/SNNservicename` script more carefully to find out how it works.

[Exercise: pick one of the regular services found with chkconfig that you're not familiar with, and locate its executable file.]

/etc/xinetd.d/fisa-kpoprelay

- Look for the "server =" line

```
service pop3
{
        disable = no
        id              = fisa-kpoprelay
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/kpoprelay
        server_args     = -l
        only_from       = 127.0.0.0/8
        bind            = 127.0.0.1
}
```

Services controlled by xinetd have their scripts in /etc/xinetd.d. The script name is the name of the service as described in chkconfig. Look for the "service" section for the line beginning "server ="; that will give the path to the actual executable.

[Exercise: pick one of the xinetd services found on your machine that you're not familiar with, and locate its executable file.]

What is this executable?

- Try `rpm -qf filename`
  - Identifies the RPM package (if any) that the file belongs to.
  - Then use `rpm -qi packagename` to display the package information
- Locate the executable with `whereis filename`; if necessary, parse with `strings filename` for clues

Okay, we've found the executable. Now what? Well, the filename can give us more information. The rpm -qf filename command will tell us if the file was installed with RPM. If so, then we can find out a lot of information with rpm -qi once we know the packagename. (Of course, we can also be lazy and do it all in one step with rpm -qfi filename if we prefer.)

If that doesn't work, we can use strings filename | more to display any readable text in the program file; that may give us clues as to what the program is. (If you're really lucky, you'll get the name of the programmer.)

[Exercise: pick one of the executable files you just identified. Use the command `rpm -qf filename` to figure out which RPM package it belongs with, then get the full information for that package. Finally, use the `strings` command to see what text might be inside that file. Is there a copyright notice? A programmer or company name?]

What's Listening?

- As root, do
  `netstat -tap > listening.services`
  `less listening.services`

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 localhost.localdom:pop3 *:*                     LISTEN
708/xinetd
tcp        0      0 *:x11                   *:*                     LISTEN
1093/X
tcp        0      0 *:ssh                   *:*                     LISTEN
696/sshd
tcp        0      0 localhost.localdom:smtp *:*                     LISTEN
841/master
tcp        0      0 localhos:x11-ssh-offset *:*                     LISTEN
3866/sshd
tcp        0      0 du139-205.aitlabs.i:ssh mommy.ait.iastate:37882 ESTABLISHED
3864/sshd
```

The other thing we need to be concerned about is services listening on our network connections. Unneeded services monitoring TCP/IP ports are security problems waiting to happen, and remote access trojans will open a port to wait for commands from whoever's running your machine. The netstat command can be used to see what processes are listening to the network; we can then use the procedures above to identify and verify the processes.

[Exercise: use `netstat -tap` to list listening services on your machine. Match them to the items in `chkconfig –list`. How are they launched?]

Services You Don't Want

- NFS and related services: nfsd, lockd, mountd, statd, portmapper, etc.
  - don't use on the Internet!
- r* services: rsh, rlogin, rexec, rcp, etc.
  - INSECURE! DON'T USE!
  - based on "trusted host" concept, but IP numbers can be spoofed
- inetd
  - Mostly older distros, replace with xinetd

There are many services that you don't want to use anymore. The NFS file system isn't especially secure; there are other ways of sharing files that are better (especially if you have AFS space available). The r* services were based on the idea of “trusted hosts”, that is, I have a list of machines (by IP number) that I know are OK, so any attempt to login, execute commands, etc. from those machines should be accepted without authentication. Unfortunately, IP addresses can be faked so this system shouldn't be used. Finally, the inetd system has been replaced by the extended inetd service (i.e., xinetd) which is more secure.

More Services to Avoid

- telnetd: use sshd instead
- ftpd: use scp and sftp instead, or use only on a dedicated ftp server that can be monitored and secured
- BIND (named), DNS server packages
  - only for an authoritative name server for a domain, shouldn't be necessary on campus

The telnet and ftp services, once the mainstay of network communication, send your password in plain text on the network, so they should be replaced with ssh, scp and sftp instead (they use SSL encryption and are more – but not perfectly – secure). If you do want to make files available anonymously use an ftp server, but only on a machine that doesn't run anything else that you can secure and that you monitor on a regular basis.

Unless you're running an isolated network, you shouldn't run a name server; we have fine DNS servers on campus and it's easy to get your machines registered with us.

Services to Avoid, Page 3

- Mail transport agents: sendmail, exim, postfix, qmail
  - unless your machine will be acting as a mail server, you don't need these; most UNIX mail clients can interact with POP3 and SMTP servers directly (though not often Kerberized POP3) and we have them readily available on campus

Mail services (especially sendmail) are historically one of the most popular security exploits. Unless you really want to run a mail server (say, for a department) it's not necessary to run an outbound mail transport agent like sendmail or postfix. Most modern mail clients (including Ximian Evolution, Kmail, Mozilla Mail/Thunderbird and even Pine) can be configured to use another SMTP server (like mailhub.iastate.edu). And for incoming mail, running kfetchmail to retrieve mail from the POP servers is quite secure; it polls the server and doesn't need to acccept incoming connections.

Disabling Services

- Use

  `/sbin/chkconfig` *`servicename`* `off`

  - Advantage: you can use `--level 2345` to disable it in multiple runlevels
- OR rename the *`Snnservicename`* scripts to *`Knnservicename`*
- Use `/etc/rc.d/init.d/`*`servicename`* `stop` to stop the service
- If necessary, take down the process with `kill PID`

[Exercise: shut down the sshd service on your machine in runlevels 3, 4 and 5.

```
/sbin/chkconfig --level 345 sshd off
/etc/rc.d/init.d/sshd stop
```

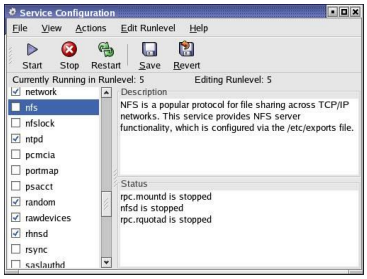
Of course, if this is your desktop machine and not a lab machine, you may want to reenable it with:

```
/sbin/chkconfig --level 345 sshd off
/etc/rc.d/init.d/sshd stop
```

You shouldn't need to kill the process; sshd is well behaved.]

Using system-config-services

- This GUI frontend makes it easy to start, stop and disable services
- To enable at runlevel startup, check the box
- Use Start and Stop buttons to start and stop the service
- Don't forget to click Save!

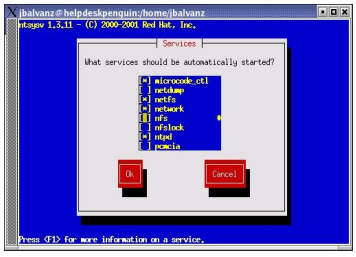


If you have access to the machine through a GUI, there's a simpler way to do all this: system-config-services. Use Start Application -> System Settings -> Server Settings -> Services to start this from the menu (you'll need to enter the root password). From here you can get info about services by clicking on the name, then start and stop with the toolbar buttons and enable and disable just by checking the box.

[Exercise: disable sshd, shut it down, then reenable it again.]

Using ntsysv

- `/usr/sbin/ntsysv` acts like system-config-services at a text console
- Use {Tab} to move cursor, {Space} to click, {F1} to see description



The ntsysv utility acts sort of like system-config-services, but doesn't need X. You can run this remotely over ssh (just don't shut down sshd) or from a text terminal. Use the Tab key to move the cursor (if you have the gpm service running you can also use the mouse), press {Space} to click. Pressing {F1} will display the same description that system-config-services does.

Exercises

- Get a list of services running on your machine
- Pick a service (preferably one you haven't seen) and find the executable that starts it
- Identify the service; what is it?
- How would you shut down the service? Restart it? Without X? With Red Hat?
